Part_I_Prsoper Loan Data

November 8, 2022

1 Part I - (Prsoper Loan Data)

1.1 by (Awaji-kansan Obediah Iduinung)

1.2 Introduction

I will be exploring the Prosper Loan Dataset. There are 113,937 loans in this dataset, and each loan has 81 variables.

1.3 Preliminary Wrangling

In [125]:
```
# import packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
%matplotlib inline
```

In [126]:
```
#Load data
LoanData = pd.read_csv('prosperLoanData.csv')
LoanData.head(5)
```

Out[126]:
```
                ListingKey  ListingNumber            ListingCreationDate  \
0  1021339766868145413AB3B         193129  2007-08-26 19:09:29.263000000
1  10273602499503308B223C1        1209647  2014-02-27 08:28:07.900000000
2  0EE9337825851032864889A          81716  2007-01-05 15:00:47.090000000
3  0EF5356002482715299901A         658116  2012-10-22 11:02:35.010000000
4  0F023589499656230C5E3E2         909464  2013-09-14 18:38:39.097000000

  CreditGrade  Term LoanStatus           ClosedDate  BorrowerAPR  \
0           C    36  Completed  2009-08-14 00:00:00      0.16516
1         NaN    36    Current                  NaN      0.12016
2          HR    36  Completed  2009-12-17 00:00:00      0.28269
3         NaN    36    Current                  NaN      0.12528
4         NaN    36    Current                  NaN      0.24614

   BorrowerRate  LenderYield    ...     LP_ServiceFees  LP_CollectionFees  \
0        0.1580       0.1380    ...            -133.18                0.0
```

```
1        0.0920       0.0820    ...            0.00               0.0
2        0.2750       0.2400    ...          -24.20               0.0
3        0.0974       0.0874    ...         -108.01               0.0
4        0.2085       0.1985    ...          -60.27               0.0

   LP_GrossPrincipalLoss  LP_NetPrincipalLoss LP_NonPrincipalRecoverypayments  \
0                    0.0                  0.0                             0.0
1                    0.0                  0.0                             0.0
2                    0.0                  0.0                             0.0
3                    0.0                  0.0                             0.0
4                    0.0                  0.0                             0.0

   PercentFunded  Recommendations InvestmentFromFriendsCount  \
0            1.0                0                          0
1            1.0                0                          0
2            1.0                0                          0
3            1.0                0                          0
4            1.0                0                          0

  InvestmentFromFriendsAmount Investors
0                         0.0       258
1                         0.0         1
2                         0.0        41
3                         0.0       158
4                         0.0        20

[5 rows x 81 columns]
```

In [127]: `LoanData.tail()`

Out[127]:

```
                    ListingKey  ListingNumber            ListingCreationDate  \
113932  E6D9357655724827169606C         753087  2013-04-14 05:55:02.663000000
113933  E6DB353036033497292EE43         537216  2011-11-03 20:42:55.333000000
113934  E6E13596170052029692BB1        1069178  2013-12-13 05:49:12.703000000
113935  E6EB3531504622671970D9E         539056  2011-11-14 13:18:26.597000000
113936  E6ED3600409833199F711B7        1140093  2014-01-15 09:27:37.657000000

       CreditGrade  Term             LoanStatus           ClosedDate  \
113932         NaN    36                Current                  NaN
113933         NaN    36  FinalPaymentInProgress                  NaN
113934         NaN    60                Current                  NaN
113935         NaN    60              Completed  2013-08-13 00:00:00
113936         NaN    36                Current                  NaN

        BorrowerAPR  BorrowerRate  LenderYield     ...      LP_ServiceFees  \
113932      0.22354        0.1864       0.1764     ...              -75.58
113933      0.13220        0.1110       0.1010     ...              -30.05
113934      0.23984        0.2150       0.2050     ...              -16.91
```

```
113935      0.28408        0.2605       0.2505     ...           -235.05
113936      0.13189        0.1039       0.0939     ...             -1.70

        LP_CollectionFees  LP_GrossPrincipalLoss  LP_NetPrincipalLoss  \
113932                0.0                    0.0                  0.0
113933                0.0                    0.0                  0.0
113934                0.0                    0.0                  0.0
113935                0.0                    0.0                  0.0
113936                0.0                    0.0                  0.0

       LP_NonPrincipalRecoverypayments  PercentFunded  Recommendations  \
113932                             0.0            1.0                0
113933                             0.0            1.0                0
113934                             0.0            1.0                0
113935                             0.0            1.0                0
113936                             0.0            1.0                0

       InvestmentFromFriendsCount InvestmentFromFriendsAmount Investors
113932                          0                         0.0         1
113933                          0                         0.0        22
113934                          0                         0.0       119
113935                          0                         0.0       274
113936                          0                         0.0         1

[5 rows x 81 columns]
```

In [128]: LoanData.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113937 entries, 0 to 113936
Data columns (total 81 columns):
ListingKey                       113937 non-null object
ListingNumber                    113937 non-null int64
ListingCreationDate              113937 non-null object
CreditGrade                      28953 non-null object
Term                             113937 non-null int64
LoanStatus                       113937 non-null object
ClosedDate                       55089 non-null object
BorrowerAPR                      113912 non-null float64
BorrowerRate                     113937 non-null float64
LenderYield                      113937 non-null float64
EstimatedEffectiveYield          84853 non-null float64
EstimatedLoss                    84853 non-null float64
EstimatedReturn                  84853 non-null float64
ProsperRating (numeric)          84853 non-null float64
ProsperRating (Alpha)            84853 non-null object
ProsperScore                     84853 non-null float64
ListingCategory (numeric)        113937 non-null int64
```

```
BorrowerState                        108422 non-null object
Occupation                           110349 non-null object
EmploymentStatus                     111682 non-null object
EmploymentStatusDuration             106312 non-null float64
IsBorrowerHomeowner                  113937 non-null bool
CurrentlyInGroup                     113937 non-null bool
GroupKey                             13341 non-null object
DateCreditPulled                     113937 non-null object
CreditScoreRangeLower                113346 non-null float64
CreditScoreRangeUpper                113346 non-null float64
FirstRecordedCreditLine              113240 non-null object
CurrentCreditLines                   106333 non-null float64
OpenCreditLines                      106333 non-null float64
TotalCreditLinespast7years           113240 non-null float64
OpenRevolvingAccounts                113937 non-null int64
OpenRevolvingMonthlyPayment          113937 non-null float64
InquiriesLast6Months                 113240 non-null float64
TotalInquiries                       112778 non-null float64
CurrentDelinquencies                 113240 non-null float64
AmountDelinquent                     106315 non-null float64
DelinquenciesLast7Years              112947 non-null float64
PublicRecordsLast10Years             113240 non-null float64
PublicRecordsLast12Months            106333 non-null float64
RevolvingCreditBalance               106333 non-null float64
BankcardUtilization                  106333 non-null float64
AvailableBankcardCredit              106393 non-null float64
TotalTrades                          106393 non-null float64
TradesNeverDelinquent (percentage)   106393 non-null float64
TradesOpenedLast6Months              106393 non-null float64
DebtToIncomeRatio                    105383 non-null float64
IncomeRange                          113937 non-null object
IncomeVerifiable                     113937 non-null bool
StatedMonthlyIncome                  113937 non-null float64
LoanKey                              113937 non-null object
TotalProsperLoans                    22085 non-null float64
TotalProsperPaymentsBilled           22085 non-null float64
OnTimeProsperPayments                22085 non-null float64
ProsperPaymentsLessThanOneMonthLate  22085 non-null float64
ProsperPaymentsOneMonthPlusLate      22085 non-null float64
ProsperPrincipalBorrowed             22085 non-null float64
ProsperPrincipalOutstanding          22085 non-null float64
ScorexChangeAtTimeOfListing          18928 non-null float64
LoanCurrentDaysDelinquent            113937 non-null int64
LoanFirstDefaultedCycleNumber        16952 non-null float64
LoanMonthsSinceOrigination           113937 non-null int64
LoanNumber                           113937 non-null int64
LoanOriginalAmount                   113937 non-null int64
LoanOriginationDate                  113937 non-null object
```

```
LoanOriginationQuarter                113937 non-null object
MemberKey                             113937 non-null object
MonthlyLoanPayment                    113937 non-null float64
LP_CustomerPayments                   113937 non-null float64
LP_CustomerPrincipalPayments          113937 non-null float64
LP_InterestandFees                    113937 non-null float64
LP_ServiceFees                        113937 non-null float64
LP_CollectionFees                     113937 non-null float64
LP_GrossPrincipalLoss                 113937 non-null float64
LP_NetPrincipalLoss                   113937 non-null float64
LP_NonPrincipalRecoverypayments       113937 non-null float64
PercentFunded                         113937 non-null float64
Recommendations                       113937 non-null int64
InvestmentFromFriendsCount            113937 non-null int64
InvestmentFromFriendsAmount           113937 non-null float64
Investors                             113937 non-null int64
dtypes: bool(3), float64(50), int64(11), object(17)
memory usage: 68.1+ MB
```

In [129]: LoanData.shape

Out[129]: (113937, 81)

In [130]: LoanData.describe()

Out[130]:

```
       ListingNumber           Term    BorrowerAPR   BorrowerRate  \
count   1.139370e+05  113937.000000  113912.000000  113937.000000
mean    6.278857e+05      40.830248       0.218828       0.192764
std     3.280762e+05      10.436212       0.080364       0.074818
min     4.000000e+00      12.000000       0.006530       0.000000
25%     4.009190e+05      36.000000       0.156290       0.134000
50%     6.005540e+05      36.000000       0.209760       0.184000
75%     8.926340e+05      36.000000       0.283810       0.250000
max     1.255725e+06      60.000000       0.512290       0.497500

         LenderYield  EstimatedEffectiveYield  EstimatedLoss  EstimatedReturn  \
count  113937.000000             84853.000000   84853.000000     84853.000000
mean        0.182701                 0.168661       0.080306         0.096068
std         0.074516                 0.068467       0.046764         0.030403
min        -0.010000                -0.182700       0.004900        -0.182700
25%         0.124200                 0.115670       0.042400         0.074080
50%         0.173000                 0.161500       0.072400         0.091700
75%         0.240000                 0.224300       0.112000         0.116600
max         0.492500                 0.319900       0.366000         0.283700

       ProsperRating (numeric)  ProsperScore      ...        LP_ServiceFees  \
count             84853.000000  84853.000000      ...         113937.000000
mean                  4.072243      5.950067      ...            -54.725641
```

```
std                1.673227      2.376501      ...          60.675425
min                1.000000      1.000000      ...        -664.870000
25%                3.000000      4.000000      ...         -73.180000
50%                4.000000      6.000000      ...         -34.440000
75%                5.000000      8.000000      ...         -13.920000
max                7.000000     11.000000      ...          32.060000

       LP_CollectionFees  LP_GrossPrincipalLoss  LP_NetPrincipalLoss  \
count      113937.000000          113937.000000        113937.000000
mean          -14.242698             700.446342           681.420499
std           109.232758            2388.513831          2357.167068
min         -9274.750000             -94.200000          -954.550000
25%             0.000000               0.000000             0.000000
50%             0.000000               0.000000             0.000000
75%             0.000000               0.000000             0.000000
max             0.000000           25000.000000         25000.000000

       LP_NonPrincipalRecoverypayments  PercentFunded  Recommendations  \
count                    113937.000000  113937.000000    113937.000000
mean                         25.142686       0.998584         0.048027
std                         275.657937       0.017919         0.332353
min                           0.000000       0.700000         0.000000
25%                           0.000000       1.000000         0.000000
50%                           0.000000       1.000000         0.000000
75%                           0.000000       1.000000         0.000000
max                       21117.900000       1.012500        39.000000

       InvestmentFromFriendsCount  InvestmentFromFriendsAmount      Investors
count               113937.000000                113937.000000  113937.000000
mean                     0.023460                    16.550751      80.475228
std                      0.232412                   294.545422     103.239020
min                      0.000000                     0.000000       1.000000
25%                      0.000000                     0.000000       2.000000
50%                      0.000000                     0.000000      44.000000
75%                      0.000000                     0.000000     115.000000
max                     33.000000                 25000.000000    1189.000000

[8 rows x 61 columns]
```

In [131]:
```
LoanData.sample(10)
```

Out[131]:
```
                     ListingKey  ListingNumber            ListingCreationDate  \
102693  CCF7337771896757382F137          80052  2007-01-01 16:24:02.657000000
94116   F5213601997294460C0F76B        1173231  2014-01-30 14:52:52.637000000
96766   3F603602033612663354 8AB       1210953  2014-02-15 16:16:26.613000000
38073   118935764272987413C4175         755752  2013-04-17 14:12:29.627000000
96666   75B93588952546041EB3A5B         916064  2013-09-16 18:40:07.470000000
85613   685A3588234991523F39EB2         882021  2013-08-28 07:32:59.207000000
```

```
41076   74073402332724401FE2495         216919  2007-10-16 16:47:13.607000000
10113   EC4835798407399963EF442         787818  2013-05-23 09:22:16.343000000
86191   51E03559553624510DAB7F8         646590  2012-09-27 23:23:56.270000000
76137   17853529195651826808B6F         535415  2011-10-26 05:00:54.523000000

       CreditGrade  Term  LoanStatus           ClosedDate  BorrowerAPR  \
102693           D    36   Completed  2008-02-20 00:00:00      0.13705
94116          NaN    60     Current                  NaN      0.13636
96766          NaN    36     Current                  NaN      0.17649
38073          NaN    36     Current                  NaN      0.32538
96666          NaN    36     Current                  NaN      0.35356
85613          NaN    36     Current                  NaN      0.18725
41076            C    36   Completed  2010-10-25 00:00:00      0.15713
10113          NaN    36     Current                  NaN      0.17192
86191          NaN    36     Current                  NaN      0.09736
76137          NaN    36  Chargedoff  2013-01-30 00:00:00      0.25486

        BorrowerRate  LenderYield    ...     LP_ServiceFees  \
102693        0.1300       0.1100    ...             -10.85
94116         0.1139       0.1039    ...             -15.34
96766         0.1400       0.1300    ...               0.00
38073         0.2859       0.2759    ...             -30.67
96666         0.3134       0.3034    ...             -17.40
85613         0.1509       0.1409    ...             -70.63
41076         0.1500       0.1400    ...             -72.67
10113         0.1359       0.1259    ...             -82.04
86191         0.0839       0.0739    ...            -281.82
76137         0.2205       0.2105    ...            -113.54

        LP_CollectionFees  LP_GrossPrincipalLoss  LP_NetPrincipalLoss  \
102693                0.0                   0.00                 0.00
94116                 0.0                   0.00                 0.00
96766                 0.0                   0.00                 0.00
38073                 0.0                   0.00                 0.00
96666                 0.0                   0.00                 0.00
85613                 0.0                   0.00                 0.00
41076                 0.0                   0.00                 0.00
10113                 0.0                   0.00                 0.00
86191                 0.0                   0.00                 0.00
76137                 0.0               11771.12             11771.12

       LP_NonPrincipalRecoverypayments  PercentFunded  Recommendations  \
102693                             0.0            1.0                0
94116                              0.0            1.0                0
96766                              0.0            1.0                0
38073                              0.0            1.0                0
96666                              0.0            1.0                0
85613                              0.0            1.0                0
```

```
41076                              0.0            1.0                0
10113                              0.0            1.0                0
86191                              0.0            1.0                0
76137                              0.0            1.0                0

       InvestmentFromFriendsCount InvestmentFromFriendsAmount Investors
102693                          0                         0.0       110
94116                           0                         0.0         1
96766                           0                         0.0         1
38073                           0                         0.0        74
96666                           0                         0.0        54
85613                           0                         0.0        62
41076                           0                         0.0       109
10113                           0                         0.0         1
86191                           0                         0.0       446
76137                           0                         0.0        79

[10 rows x 81 columns]
```

In [132]:
```
#Select columns to explore
needed_columns = ['Term', 'ListingCategory (numeric)', 'CreditGrade', 'EstimatedReturn
```

In [133]:
```
Target_LoanData = LoanData[needed_columns]
```

In [134]:
```
Target_LoanData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113937 entries, 0 to 113936
Data columns (total 17 columns):
Term                         113937 non-null int64
ListingCategory (numeric)    113937 non-null int64
CreditGrade                  28953 non-null object
EstimatedReturn              84853 non-null float64
Investors                    113937 non-null int64
StatedMonthlyIncome          113937 non-null float64
AmountDelinquent             106315 non-null float64
ProsperScore                 84853 non-null float64
LoanOriginalAmount           113937 non-null int64
MonthlyLoanPayment           113937 non-null float64
LoanStatus                   113937 non-null object
BorrowerRate                 113937 non-null float64
ProsperRating (Alpha)        84853 non-null object
LoanOriginationDate          113937 non-null object
EmploymentStatus             111682 non-null object
Occupation                   110349 non-null object
IncomeRange                  113937 non-null object
dtypes: float64(6), int64(4), object(7)
memory usage: 14.8+ MB
```

```
In [135]: #Drop missing values in the ProsperRating (Alpha), AmountDelinquent, ProsperScore, Emp
          Target_LoanData = Target_LoanData.dropna(subset=['ProsperRating (Alpha)', 'AmountDelin

In [136]: #Convert datatype of LoanOriginationDate to datetime
          Target_LoanData['LoanOriginationDate'] = pd.to_datetime(Target_LoanData['LoanOriginati

In [137]: Target_LoanData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 83520 entries, 0 to 83519
Data columns (total 18 columns):
index                        83520 non-null int64
Term                         83520 non-null int64
ListingCategory (numeric)    83520 non-null int64
CreditGrade                  0 non-null object
EstimatedReturn              83520 non-null float64
Investors                    83520 non-null int64
StatedMonthlyIncome          83520 non-null float64
AmountDelinquent             83520 non-null float64
ProsperScore                 83520 non-null float64
LoanOriginalAmount           83520 non-null int64
MonthlyLoanPayment           83520 non-null float64
LoanStatus                   83520 non-null object
BorrowerRate                 83520 non-null float64
ProsperRating (Alpha)        83520 non-null object
LoanOriginationDate          83520 non-null datetime64[ns]
EmploymentStatus             83520 non-null object
Occupation                   83520 non-null object
IncomeRange                  83520 non-null object
dtypes: datetime64[ns](1), float64(6), int64(5), object(6)
memory usage: 11.5+ MB
```

```
In [138]: Target_LoanData['LoanOriginationDate']
```

```
Out[138]: 0       2014-03-03
          1       2012-11-01
          2       2013-09-20
          3       2013-12-24
          4       2013-04-18
          5       2013-05-13
          6       2013-12-12
          7       2013-12-12
          8       2012-05-17
          9       2014-01-07
          10      2013-07-18
          11      2013-05-13
          12      2012-04-19
          13      2013-07-18
```

```
14       2013-03-11
15       2013-10-10
16       2013-11-29
17       2013-02-05
18       2013-04-26
19       2013-12-18
20       2013-10-10
21       2013-02-21
22       2010-06-24
23       2013-11-13
24       2014-01-16
25       2012-02-07
26       2012-09-27
27       2014-01-22
28       2010-10-26
29       2011-12-21
            ...
83490   2009-12-28
83491   2013-12-16
83492   2010-04-09
83493   2010-03-18
83494   2014-03-03
83495   2013-11-26
83496   2014-02-28
83497   2011-12-05
83498   2013-11-13
83499   2010-12-08
83500   2012-09-17
83501   2014-01-29
83502   2013-11-27
83503   2013-12-20
83504   2010-05-05
83505   2012-11-28
83506   2013-11-29
83507   2013-05-14
83508   2013-06-13
83509   2012-10-23
83510   2013-05-08
83511   2011-06-10
83512   2013-07-10
83513   2013-07-10
83514   2014-01-22
83515   2013-04-22
83516   2011-11-07
83517   2013-12-23
83518   2011-11-21
83519   2014-01-21
Name: LoanOriginationDate, Length: 83520, dtype: datetime64[ns]
```

1.3.1 What is the structure of your dataset?

There are 113937 rows and 81 colomuns in the dataset

1.3.2 What is/are the main feature(s) of interest in your dataset?

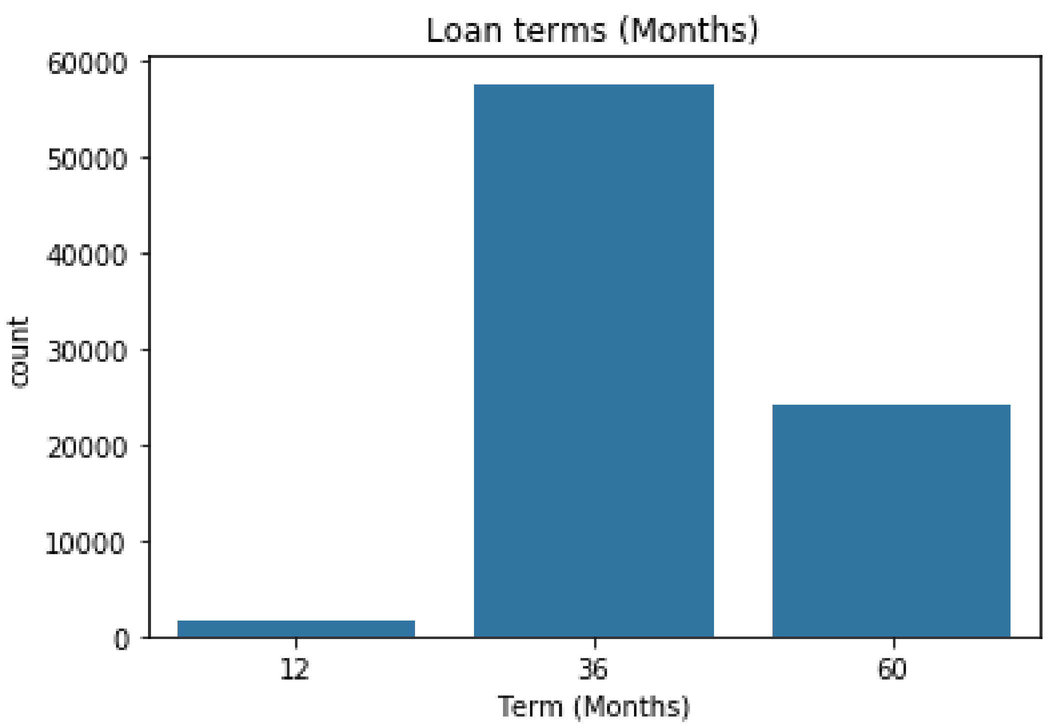
My interest is analyse the relationship between Prosper rating, emplyment status, loan status, laon amount, and the duration of the loan

1.3.3 What features in the dataset do you think will help support your investigation into your feature(s) of interest?

Term, StatedMonthlyIncome, ProsperScore, LoanOriginalAmount, MonthlyLoanPayment, LoanStatus, ProsperRating (Alpha), ListingCategory (numeric), and EmploymentStatus are the features I will be considering in this investigation.

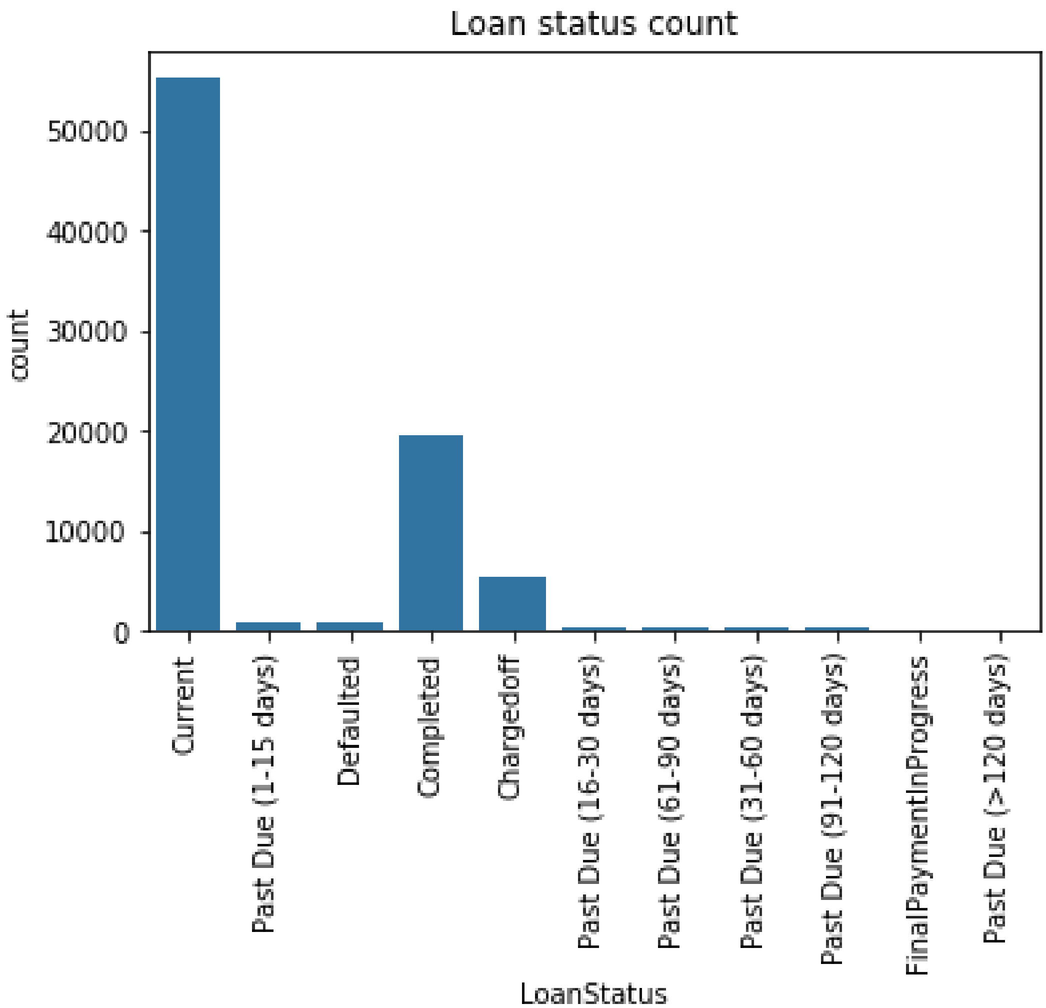
1.4 Univariate Exploration

In [139]:
```
# Visualise the loan term
base_color = sb.color_palette()[0]
sb.countplot(x ='Term', data = Target_LoanData, color=base_color);
plt.title('Loan terms (Months)')
plt.xlabel('Term (Months)');
```



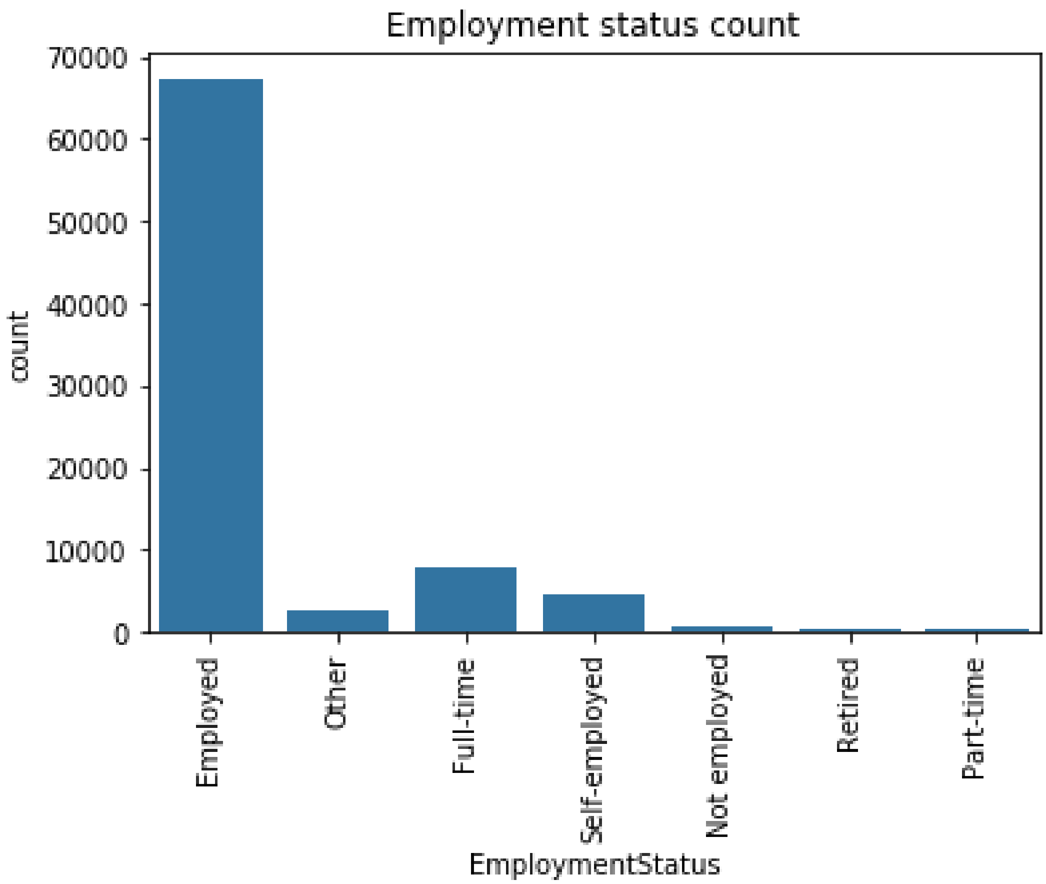
Majority of the loans have a legnth of 36 months

In [140]:
```
# Visualise the loan status
base_color = sb.color_palette()[0]
plt.xticks(rotation=90)
sb.countplot(x ='LoanStatus', data = Target_LoanData, color=base_color);
plt.title('Loan status count');
```



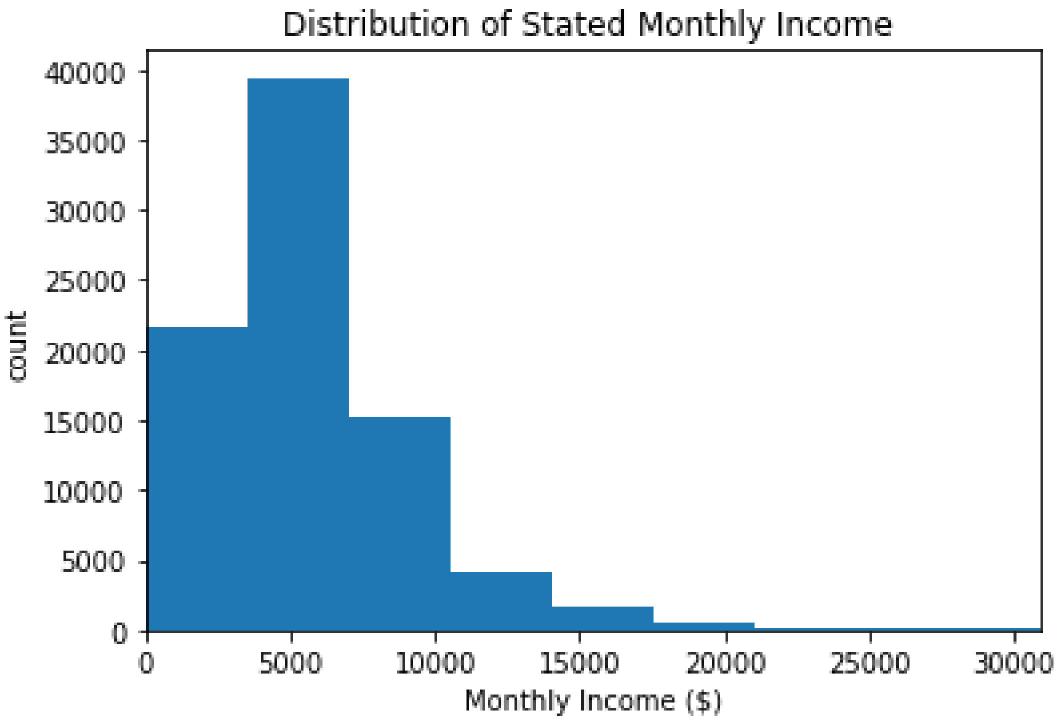
Majority of the loans are current loans, followed by completed loand and then Charged off loans

In [141]:
```
# Visualise the employment status
sb.countplot(data = Target_LoanData, x = 'EmploymentStatus', color = base_color);
plt.xticks(rotation = 90);
plt.title('Employment status count');
```



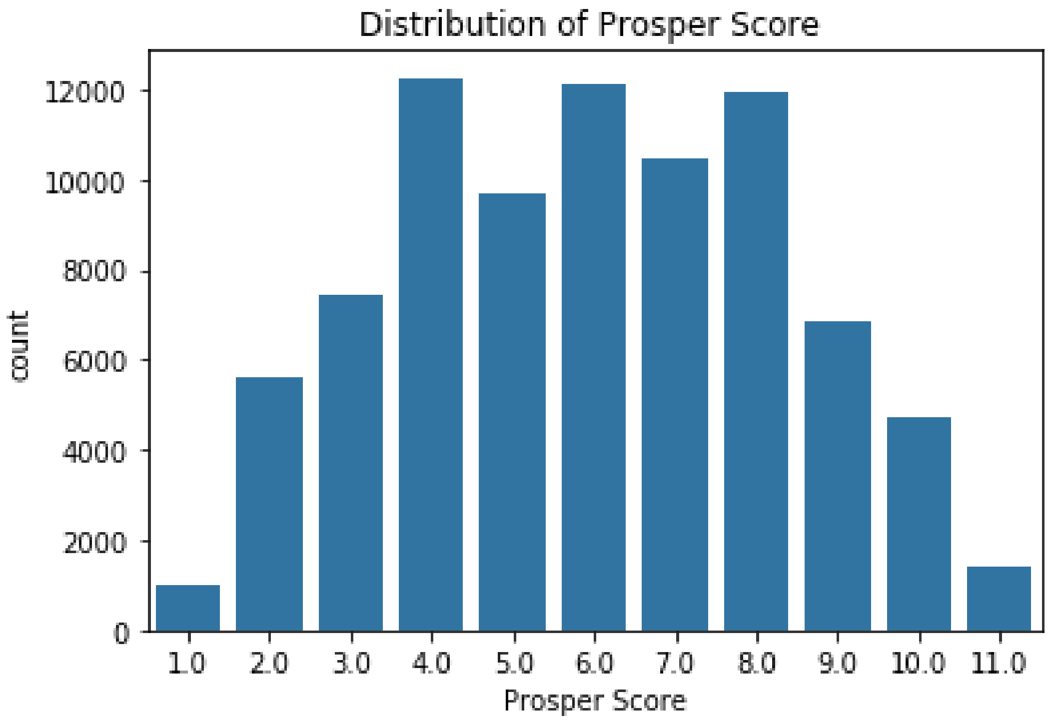
This plot shows that majority of the borrowers are employed. It also shows that those on part-time employment and retirees have the least number of loans.

```
In [142]: # Visualise the stated monthly income
          StatedMonthlyIncome_std = Target_LoanData['StatedMonthlyIncome'].std()
          StatedMonthlyIncome_mean = Target_LoanData['StatedMonthlyIncome'].mean()
          boundary = StatedMonthlyIncome_mean + StatedMonthlyIncome_std * 3
          len(Target_LoanData[Target_LoanData['StatedMonthlyIncome'] >= boundary])
          plt.hist(data=Target_LoanData, x='StatedMonthlyIncome', bins=500);


          plt.xlim(0, boundary);
          plt.title('Distribution of Stated Monthly Income ')
          plt.xlabel('Monthly Income ($)');
          plt.ylabel('count');
```



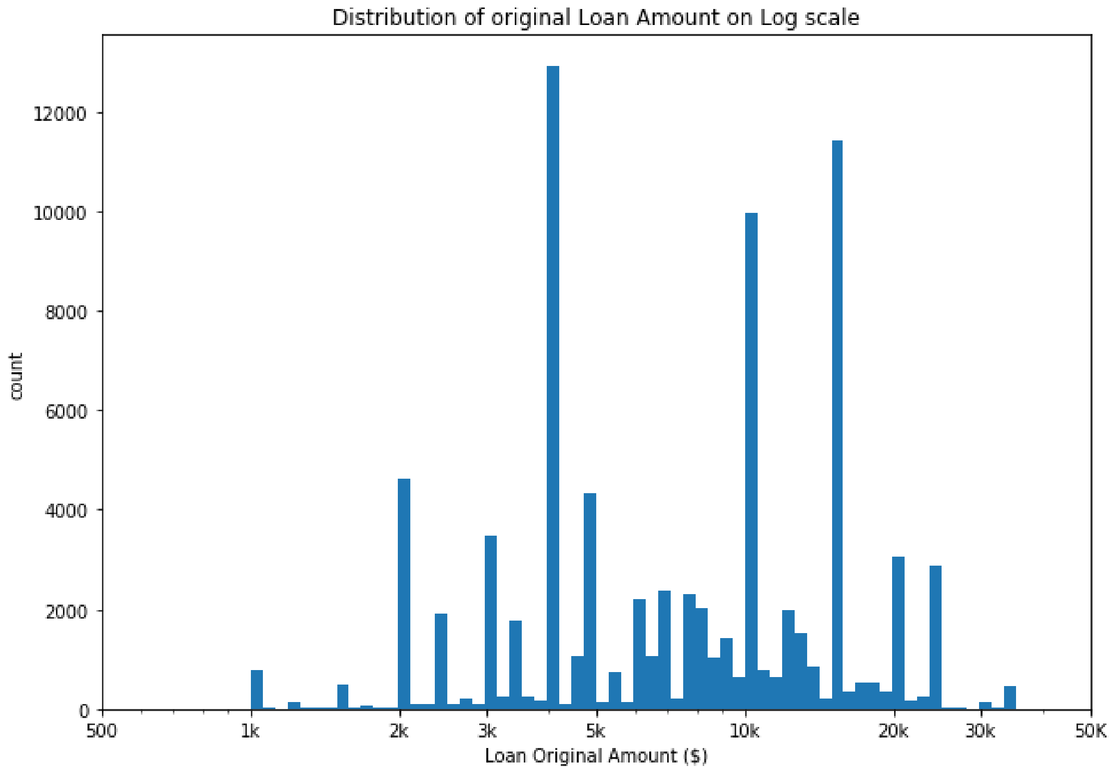
We can see from the histogram above that most of borrowers stated $5000 as their monthly. Very few of the borrowers stated that they earn $30000 monthly

```
In [143]: # Visualise the Prosper score
          base_color = sb.color_palette()[0]
          sb.countplot(data=Target_LoanData, x= 'ProsperScore', color=base_color)
          plt.title('Distribution of Prosper Score ')
          plt.xlabel('Prosper Score');
```



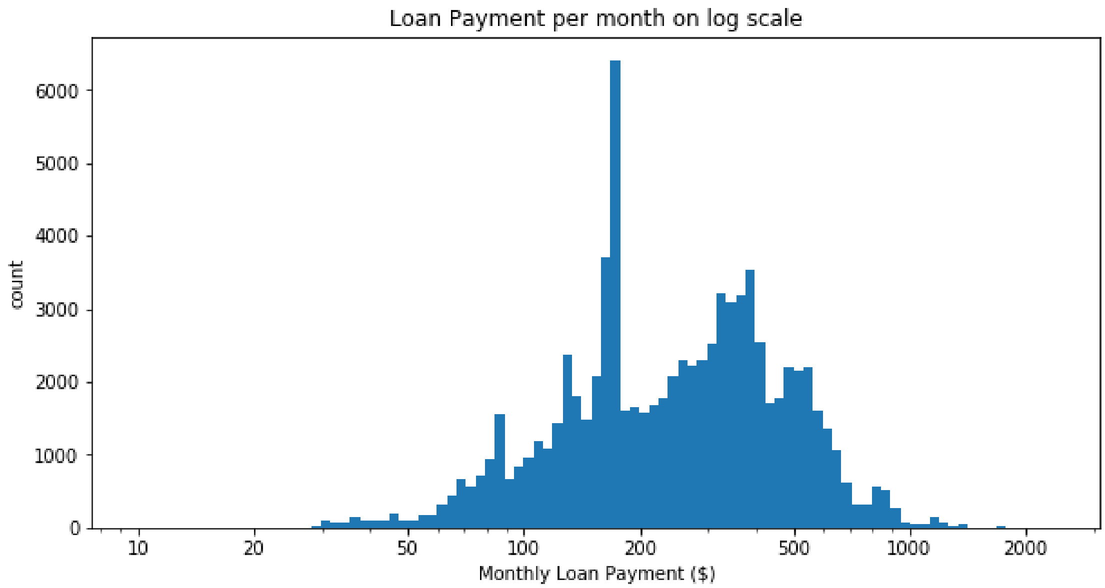
This plot shows an almost normal distribution of risk scores. While 1.0 is the lowest score, 4.0, 6.0, and 8.0 are the highest recorded scores

```
In [144]: # Visualise the loan Original Amount in a log-scale
          log_binsize = 0.025
          bins = 10 ** np.arange(3, np.log10(Target_LoanData['LoanOriginalAmount'].max())+log_bi

          plt.figure(figsize=[10, 7])
          plt.hist(data = Target_LoanData, x = 'LoanOriginalAmount', bins = bins)
          plt.title('Distribution of original Loan Amount on Log scale')
          plt.xscale('log')
          plt.xticks([500, 1e3, 2e3,3e3, 5e3, 1e4, 2e4, 3e4, 5e4], ['500', '1k', '2k', '3k', '5k
          plt.xlabel('Loan Original Amount ($)')
          plt.ylabel('count')
          plt.show()
```



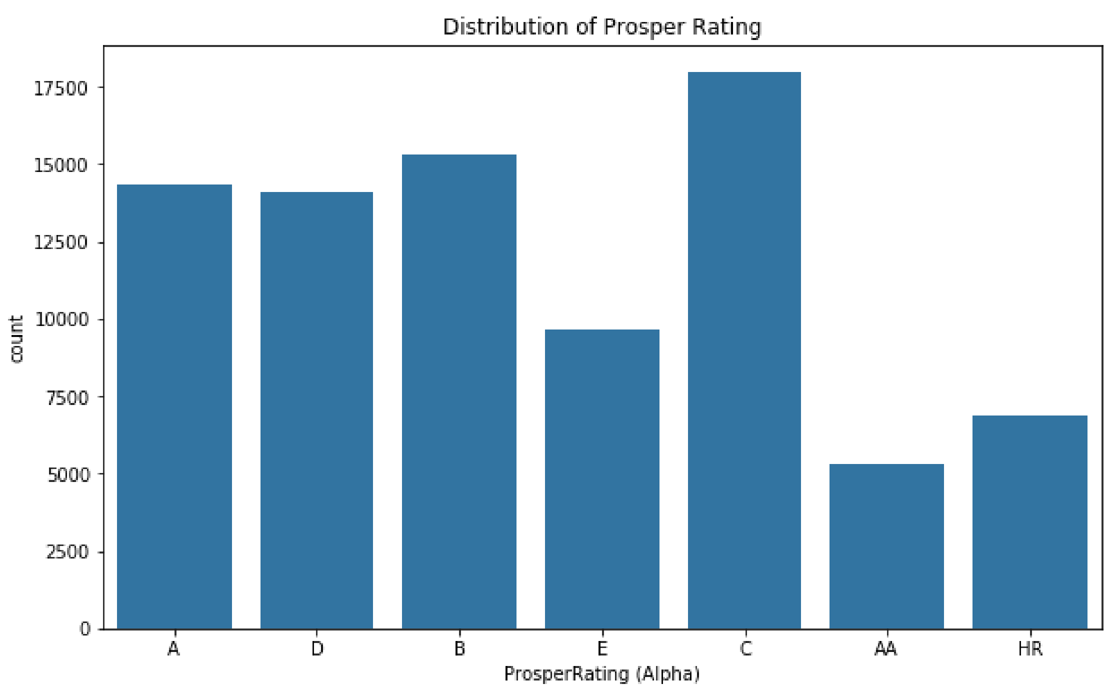
It can be deduced from the above plot most of the borrowers took a loan of about $4k, followed by loans of about $17k and $10k respectively

In [145]:
```
# Visualise the monthly loan payment using the log-scale
log_binsize = 0.025
bins = 10 ** np.arange(1, np.log10(Target_LoanData['MonthlyLoanPayment'].max())+log_bi

plt.figure(figsize=[10, 5])
plt.hist(data = Target_LoanData, x = 'MonthlyLoanPayment', bins = bins)
plt.xscale('log')
plt.xticks([10, 20, 50, 100, 200, 500, 1e3, 2e3], ['10', '20','50', '100', '200', '500
plt.xlabel('Monthly Loan Payment ($)')
plt.ylabel('count')
plt.title('Loan Payment per month on log scale')
plt.show()
```



Majority of the borrows paid between $100 and $200 per month, followed by amounts between $300 and $400 and then $500

In [146]:
```
# Visualise the loan Prosper Rating Distribution

plt.figure(figsize=[10, 6]);
sb.countplot(data=Target_LoanData,x='ProsperRating (Alpha)', color=base_color);
plt.title('Distribution of Prosper Rating');
```



Majority of the borrowers recieved a prosper rating of C while AA rating was recieved by the least number of borrowers

1.4.1 Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you need to perform any transformations?

Prosper rating is almost evenly distributed, with the highest rating being C. Emplyment and Loan status are both skewed to the left with most of the borrowers being employed and most of the loan being current loans. For the original loan amount, I visualised it on a log sclae and the distribution appears skewed to the right.

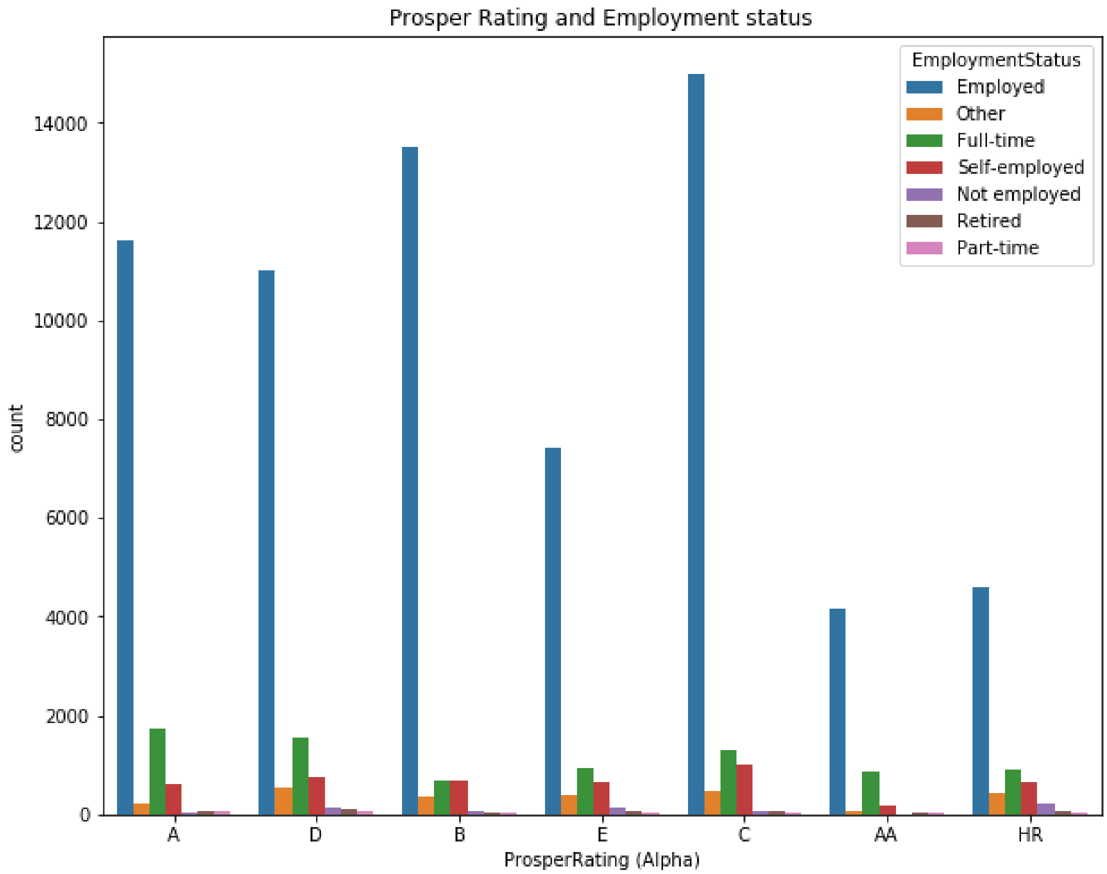
Most of the loan has a duration of 36 months, followed by 60 months duration and then 12 months.

1.4.2 Of the features you investigated, were there any unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

My main interest is ProsperRating (Alpha), AmountDelinquent, ProsperScore, EmploymentStatus which had missin values so I went ahead to drop the missing values in these variables.

1.5 Bivariate Exploration

```
In [147]: # Visualise the prosper rating vs the employment status
          plt.figure(figsize = [10, 8])
          sb.countplot(x ='ProsperRating (Alpha)', hue = 'EmploymentStatus', data = Target_LoanD
          plt.title('Prosper Rating and Employment status');
```



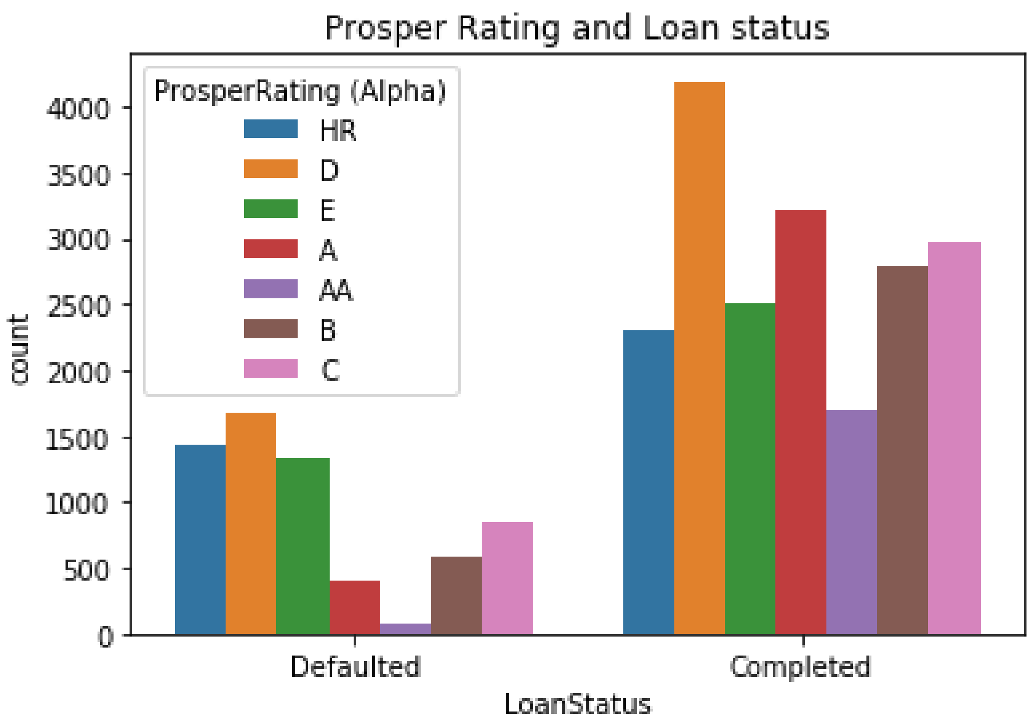
Employed people appear the most in all the rating categories. Most of the employed people recieved a D rating and the least number of employed people have the highest rating of AA. This pattern can be observed for all other employemnt status across the different catrgories of ratings.

```
In [148]: condition = (Target_LoanData['LoanStatus'] == 'Completed') | (Target_LoanData['LoanSta
                             (Target_LoanData['LoanStatus'] == 'Chargedoff')
          Target_LoanData = Target_LoanData[condition]

          def change_to_defaulted(row):
              if row['LoanStatus'] == 'Chargedoff':
                  return 'Defaulted'
              else:
                  return row['LoanStatus']

          Target_LoanData['LoanStatus'] = Target_LoanData.apply(change_to_defaulted, axis=1)
          categories = {1: 'Debt Consolidation', 2: 'Home Improvement', 3: 'Business', 6: 'Auto'
          def reduce_categorie(row):
              loan_category = row['ListingCategory (numeric)']
              if  loan_category in categories:
                  return categories[loan_category]
              else:
```

```
            return categories[7]

    Target_LoanData['ListingCategory (numeric)'] = Target_LoanData.apply(reduce_categorie,
```

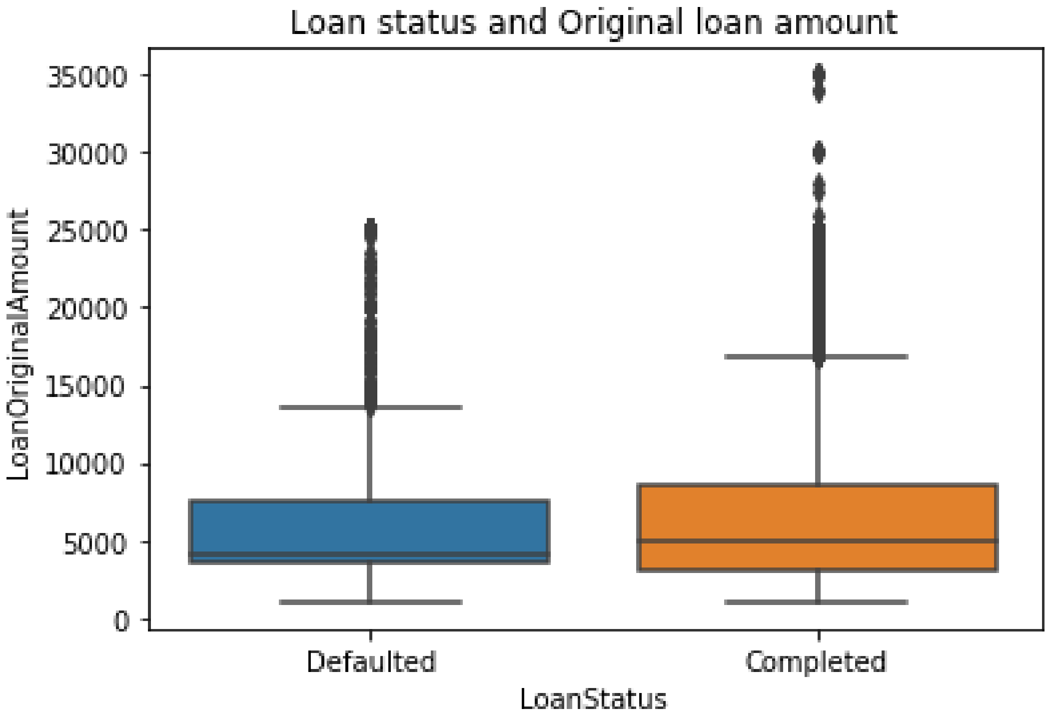
ref: https://github.com/Abhishek20182/Communicate-Data-Findings/blob/master/exploration.ipynb

```
In [149]: # Visualise theprosper rating vs loan status
          sb.countplot(x ='LoanStatus', hue = 'ProsperRating (Alpha)', data = Target_LoanData);
          plt.title('Prosper Rating and Loan status');
```



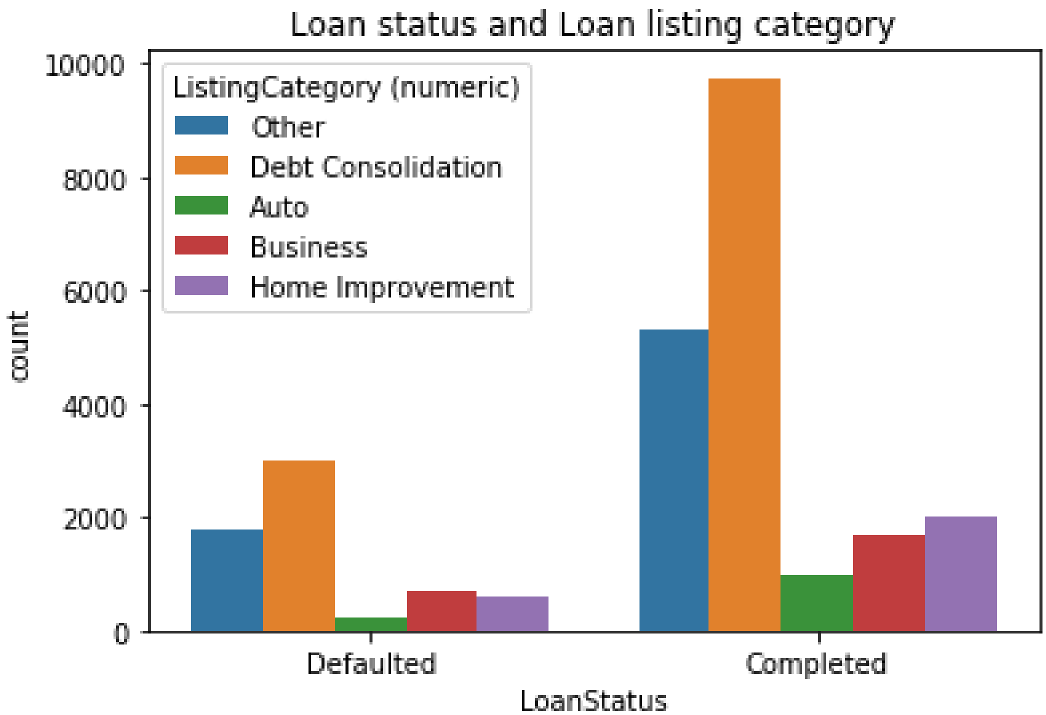
Both the defaulted and completed loans have D as the highest rating. While A rating is he second highest recorded rating for completed loans, HR rating is the second highest for defaulted loans.

```
In [150]: # Visualise the loan status and original loan amount
          sb.boxplot(data = Target_LoanData, x = 'LoanStatus', y = 'LoanOriginalAmount');
          plt.title('Loan status and Original loan amount');
```



Completed loans appear to have higher original loan amount while the loan amount for defaulted loans are lower

In [151]:
```
# Visualise the loan status vs Loan listing category
sb.countplot(x ='LoanStatus', hue = 'ListingCategory (numeric)', data = Target_LoanDat
plt.title('Loan status and Loan listing category');
```



The lowest listing category in both the defaulted and completed loans is the Auto, while Debt Consolidation has the highest frequency in both.

1.5.1 Talk about some of the relationships you observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

In Prosper Rating vs Loan status plot, the AA rating appears the least in both the defaulted and completed loan status. In 'Prosper Rating vs employment status, employed status appears the most across all the rating categories.

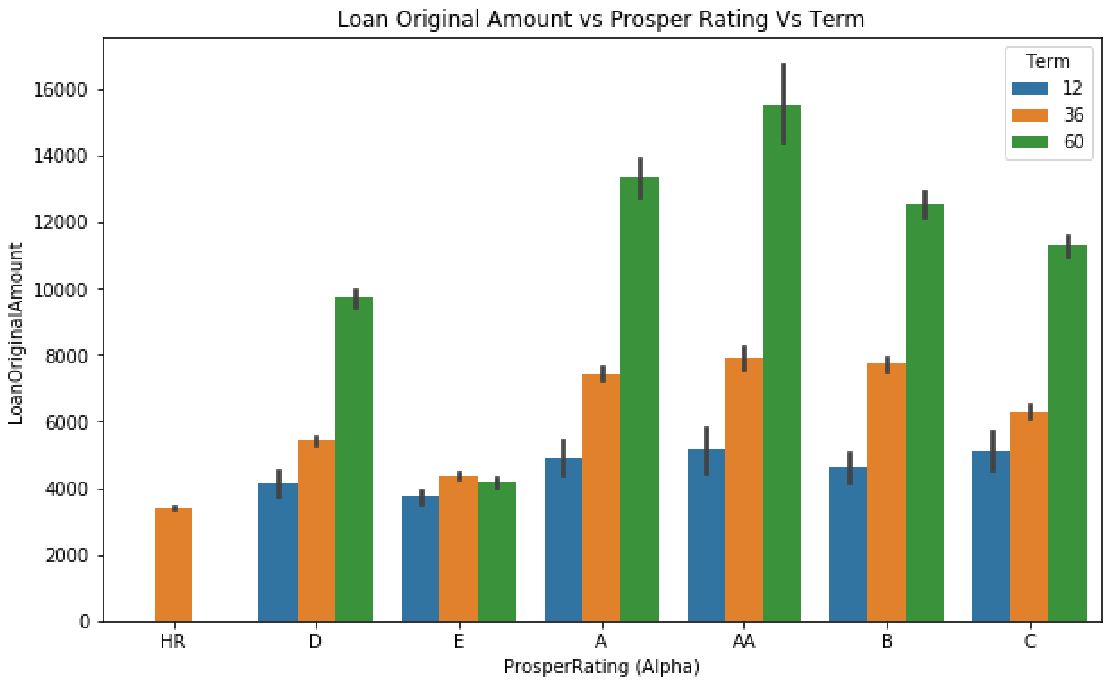
1.5.2 Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

Borrowers with the Listing Category of Debt Consolidation appear the most in both the completed and defaulted loan status.

1.6 Multivariate Exploration

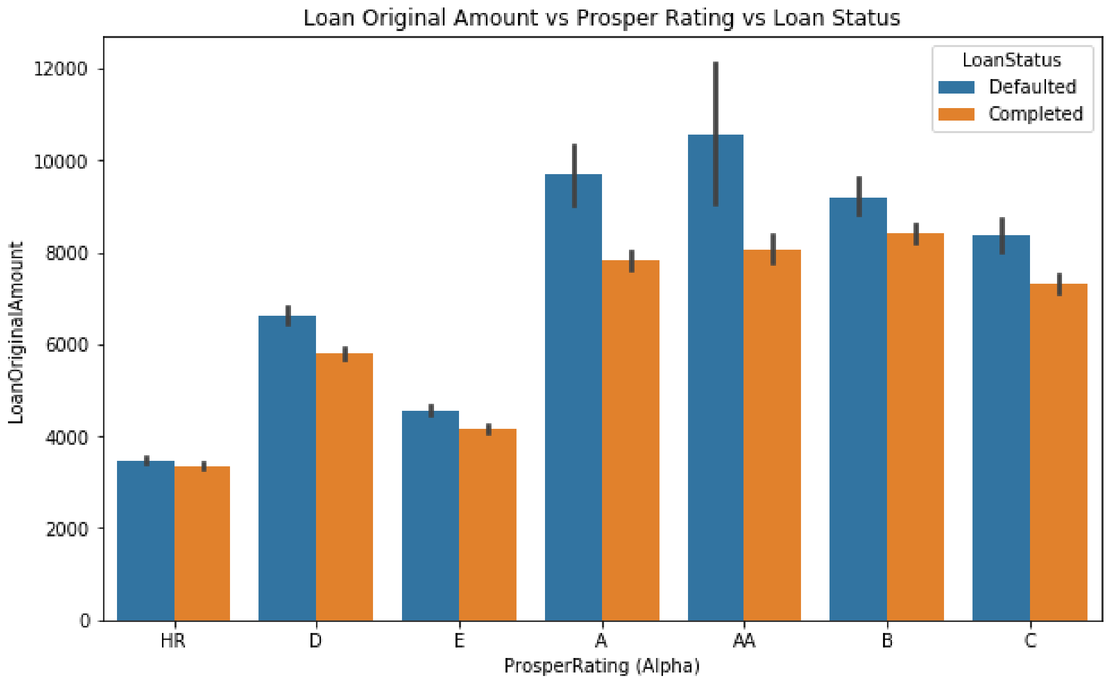
```
In [152]: # visualise the Prosper Rating, Stated Monthly Income and Term of the loan
          plt.figure(figsize = [10, 6])
          sb.barplot(
              data=Target_LoanData,
              x='ProsperRating (Alpha)',
              y='LoanOriginalAmount',
              hue='Term');
```

```
plt.title('Loan Original Amount vs Prosper Rating Vs Term');
```



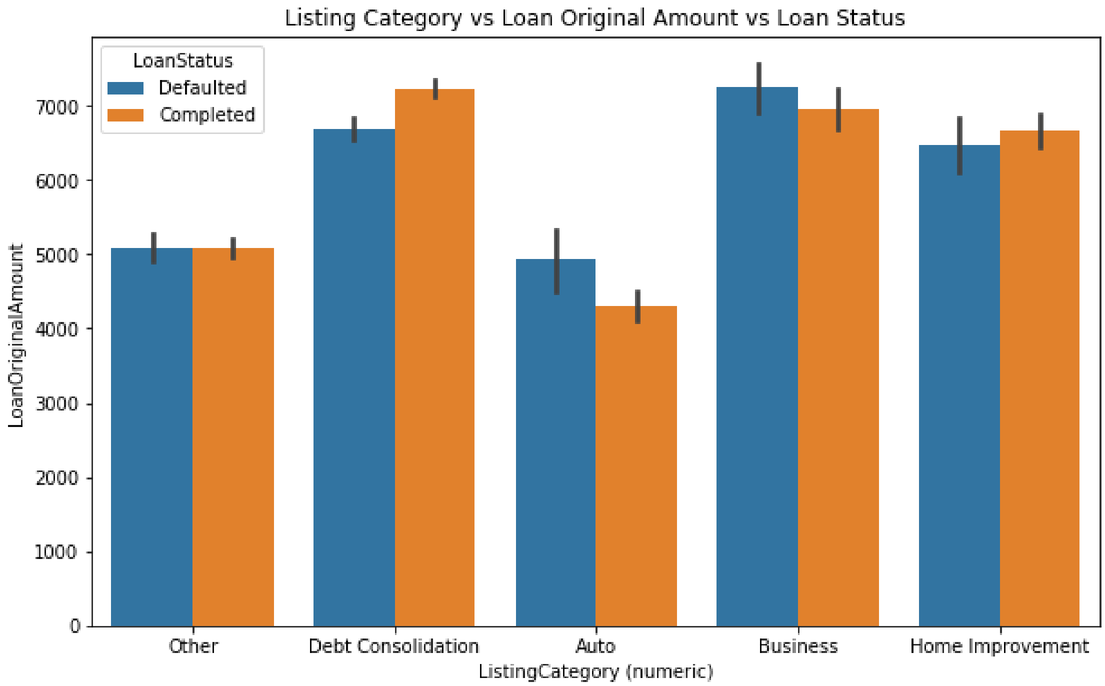
There is a clear interaction between Prosper rating, loan original amountand, and Term. At Prosper rating AA which is the highest rating, the loan amount for all three terms is highest.

```
In [153]: # visualise the Prosper Rating, Loan Original Amount and LoanStatus
          plt.figure(figsize = [10, 6])
          sb.barplot(
              x='ProsperRating (Alpha)',
              y='LoanOriginalAmount',
              data=Target_LoanData,
              hue='LoanStatus');
          plt.title('Loan Original Amount vs Prosper Rating vs Loan Status');
```



It can be observed here that defaulted loans clearly have the highest loan amount across all the prosper rating category, except in HR where the diffrence is not too much.

```
In [154]: # visualise the ListingCategory (numeric), Loan Original Amount and LoanStatus
          plt.figure(figsize = [10, 6])
          sb.barplot(
              x='ListingCategory (numeric)',
              y='LoanOriginalAmount',
              data=Target_LoanData,
              hue='LoanStatus');
          plt.title('Listing Category vs Loan Original Amount vs Loan Status');
```



The "Other" listing category has the equal loan original amount in both the default and completed status. There is a slight difference in the loan amount of the rest of the listing category.

1.6.1 Talk about some of the relationships you observed in this part of the investigation. Were there features that strengthened each other in terms of looking at your feature(s) of interest?

Borrowers with low prosper rating have most of the defaulted loans and from the Loan Original Amount vs Listing Category vs Loan Status visualisation, larger loan amounts are associated with the Business and Debt Consolidation categories.

1.6.2 Were there any interesting or surprising interactions between features?

There is a clear interaction between the prosper rating, loan aount and the term of the loan. At Prosper rating AA which is the highest rating, the loan amount for all three terms is highest.

1.7 Conclusions

My interest was to find how the the prosper rating is affected by employment status, loan status, . and how it also relates to the original loan amount and the loan term.

The investigation I carried indicated some relationships between the variables under investigation, For example, there is a direct relationship between the prosper rating, loan amount and term of the loan